UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/605,189	09/12/2003	Gregor P. Freund	VIV/0011.01	2188

28653 7590 08/19/2008

JOHN A. SMART
201 LOS GATOS
SARATOGA RD, #161
LOS GATOS, CA 95030-5308

EXAMINER	
TRUVAN, LEYNNA THANH	
ART UNIT	**PAPER NUMBER**
2135	

MAIL DATE	DELIVERY MODE
08/19/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/605,189	FREUND, GREGOR P.	
	Examiner	Art Unit	
	Leynna T. Truvan	2135	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *23 April 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-47* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-47* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-47 are pending.

Response to Arguments

2. Applicant's arguments with respect to claims 1-47 have been considered but are moot in view of the new ground(s) of rejection.

The preamble is generally not accorded any patentable weight where it merely recites the purpose of a process or the intended use of a structure, and where the body of the claim does not depend on the preamble for completeness Innuendo that the preamble should be taken in to light with the body as the claimed invention, examiner will bring forth Johnson, et al. that an interprocess communication is under control of an operating system.

a) In response to applicant's arguments, the recitation "in a computer system operating under control of an operating system supporting interprocess communication, a method for controlling interprocess communication occurring between an application executing on the computer system and a service provided by the operating system" has not been given patentable weight because the recitation occurs in the preamble. A preamble is generally not accorded any patentable weight where it merely recites the purpose of a process or the intended use of a structure, and where the body of the claim does not depend on the preamble for completeness but, instead, the process steps or structural limitations are able to stand alone. See *In re Hirao*, 535 F.2d 67, 190

USPQ 15 (CCPA 1976) and *Kropa v. Robie*, 187 F.2d 150, 152, 88 USPQ 478, 481 (CCPA 1951).

The Ablay and Johnson inventions includes the ICP concept but if the claimed interprocess communication means something other that what is known in the art versus applicant's invention versus the Pearson reference, than it should be noted. However, Applicant acknowledges the ICP is a well-known concept, thus, shouldn't have to repeatedly use the ICP terms many times to reconfirm that Ablay's invention in fact includes the ICP concept.

In addition, by merely amending the preamble to include an operating system does not give it any more patentable weight than when it was not amended. This ICP concept is well known in the computer technology that in a computer system operating under control of an operating system supporting interprocess communication. Unless it is claimed that the computer system is processing at a particular point such as during boot-up or BIOS or runtime, then an operating system in the well known in the art computer system is needed to function or process instructions of the applications and/or services. If prior art teaches a computer system with a processor, than it will need some form of operating system. Again this is not something of a new inventive concept like the ICP concept. Examiner have cited some prior art in the PTO-892 form of these well known techniques dating all the way back to 1987 that is included in this Office Action.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1-47 are rejected under 35 U.S.C. 102(b) as anticipated by Ablay, et al. (US 6,002,941) or, in the alternative, under 35 U.S.C. 103(a) as obvious over Johnson, et al. (5,133,053).

As per claim 1:

Ablay discloses in a computer system operating under control of an operating system supporting interprocess communication, a method for controlling interprocess communication occurring between an application executing on the computer system and a service provided by the operating system, the method comprising::

defining rules **(col.8, lines 53-67)** indicating which system services of the operating system **(col.4, lines 15-23 and col.5, lines 31-37)** a given application can invoke using interprocess communication to invoke said system services; **(col.2, line 65 – col.3, lines 9 and col.6, lines 5-9)**

trapping an attempt by a particular application to invoke a particular system service; **(col.3, lines 13-22 and col.4, lines 34-52)**

identifying the particular application that is attempting to invoke the particular system service; and **(col.5, lines 62-67 and col.12, lines 48-60)**

based on identity of the particular application and on the rules indicating which system services a given application can invoke **(col.4, lines 56-65 and col.8, lines 53-**

67), blocking the attempt when the rules indicate that the particular application cannot invoke the particular system service. **(col.3, lines 1-9 and col.10, lines 5-18)**

Ablay discloses the service environment can be embodied within a stand alone computer (col.3, lines 20-28) and a Windows NT operating system was used to provide the service creation environment (col.4, lines 20-31). Ablay also discusses the computer operating system must provide interoperability between the higher-level applications and the underlying computer hardware (col.5, lines 30-35) based on rules (col.9, lines 14-40). Thus, shows the operating system of a computer system supports interprocess communication that invokes service according to an application rule in a computer system. However, a secondary art is brought forth to show the well known interprocess methodology existed wherein a computer system operating under the operating system supporting an interprocess communication.

Johnson is brought forth proving that interprocess communication may occur in the operating system (UNIX) of a computer system was known even prior to his invention. Johnsons invention facilitates interprocess communication among processes located at different nodes of a network (col.1, line 58—col.2, line 3). The interprocess communication within the UNIX system is taught by M.J. Bach in 1986 so that his invention is an improvement to Bach's. Thus, obviously shows that (Bach's) interprocess communication is supported by an operating system within a computer system rather than at different nodes of a network of Johnson's improvement.

As per claim 2: See col.3, lines 13-22 and col.4, lines 34-52; discussing the method of claim 1, wherein said trapping step includes intercepting operating system calls for

invoking the particular system service.

As per claim 3: See col.3, lines 13-22 and col.4, lines 34-52; discussing the method of claim 1, wherein said trapping step includes intercepting local procedure calls for invoking the particular system service.

As per claim 4: See col.3, lines 13-22 and col.4, lines 34-52; discussing the method of claim 1, wherein said trapping step includes intercepting an attempt to open a communication channel to the particular system service.

As per claim 5: See col.3, lines 1-9 and col.10, lines 5-18; discussing the method of claim 1, wherein said trapping step includes rerouting an attempt to invoke the particular system service from a system dispatch table to an interprocess communication controller for determining whether to block the attempt based on the rules.

As per claim 6: See col.3, lines 1-9 and col.10, lines 5-18; discussing the method of claim 5, wherein said step of rerouting attempts to invoke the particular system service from a dispatch table to the interprocess communication controller includes replacing an original destination address in the system dispatch table with an address of the interprocess communication controller.

As per claim 7: See col.5, lines 62-67 and col.8, lines 32-53; discussing the method of claim 6, further comprising the steps of: retaining the original destination address; and using the original destination address for invoking the particular system service if the interprocess communication controller determines not to block the attempt.

As per claim 8: See col.5, lines 40-55; discussing the method of claim 1, wherein the rules specifying which system services a given application can invoke are established

based on user input.

As per claim 9: See col.5, lines 40-55; discussing the method of claim 1, wherein the step of blocking the attempt is based upon consulting a rules engine for determining whether the particular application can invoke the particular system service.

As per claim 10: See col., lines ; discussing the method of claim 1, wherein the step of blocking the attempt includes obtaining user input as to whether the particular application can invoke the particular system service.

As per claim 11: See col.6, lines 1-7; discussing the method of claim 10, wherein said step of obtaining user input as to whether the particular application can invoke the particular system service includes the substeps of: providing information to the user about the particular application that is attempting to invoke the particular system service; and receiving user input as to whether the particular application should be blocked from invoking the particular system service.

As per claim 12: See col.9, lines 8-10; discussing the computer-readable medium having computer-executable instructions for performing the method of claim 1.

As per claim 13: See col.9, lines 8-10; discussing downloading a set of computer-executable instructions for performing the method of claim 1.

As per claim 14:

Ablay discloses in a computer system operating under control of an operating system supporting interprocess communication, a method for regulating communications between processes that attempt to use said interprocess communication, the method comprising:

defining a policy specifying whether one process may use interprocess communication of the operating system **(col.4, lines 15-23 and col.5, lines 31-37)** to communicate with another process; **(col.2, line 65 – col.3, lines 9 and col.6, lines 5-9)**

intercepting an attempt by a first process to communicate with a second process; **(col.3, lines 13-22 and col.4, lines 34-52)**

identifying the first process that is attempting to communicate with the second process; **(col.5, lines 62-67)**

identifying the second process; **(col.12, lines 39-46)**

based on said policy, determining whether the first process may communicate with the second process; and **(col.3, lines 1-21 and col.10, lines 5-18)**

allowing the first process to communicate with the second process if said policy indicates that the first process may communicate with the second process. **(col.5, lines 30-67 and col.12, lines 39-60)**

Ablay discloses the service environment can be embodied within a stand alone computer (col.3, lines 20-28) and a Windows NT operating system was used to provide the service creation environment (col.4, lines 20-31). Ablay also discusses the computer operating system must provide interoperability between the higher-level applications and the underlying computer hardware (col.5, lines 30-35) based on rules (col.9, lines 14-40). Thus, shows the operating system of a computer system supports interprocess communication that invokes service according to an application rule in a computer system. However, a secondary art is brought forth to show the well known interprocess

methodology existed wherein a computer system operating under the operating system supporting an interprocess communication.

Johnson is brought forth proving that interprocess communication may occur in the operating system (UNIX) of a computer system was known even prior to his invention. Johnsons invention facilitates interprocess communication among processes located at different nodes of a network (col.1, line 58—col.2, line 3). The interprocess communication within the UNIX system is taught by M.J. Bach in 1986 so that his invention is an improvement to Bach's. Thus, obviously shows that (Bach's) interprocess communication is supported by an operating system within a computer system rather than at different nodes of a network of Johnson's improvement.

As per claim 15: See col.5, lines 30-67 and col.12, lines 39-60; discussing the method of claim 14, wherein the first process comprises an instance of an application program.

As per claim 16: See col.5, lines 30-67 and col.12, lines 39-60; discussing the method of claim 14, wherein the second process comprises a system service.

As per claim 17: See col.5, lines 30-67 and col.12, lines 39-60; discussing the method of claim 14, wherein said intercepting step includes intercepting operating system calls made by the first process to attempt to communicate with the second process.

As per claim 18: See col.5, lines 30-67 and col.12, lines 39-60; discussing the method of claim 14, wherein said intercepting step includes detecting local procedure calls.

As per claim 19: See col.5, lines 30-67 and col.12, lines 39-60; discussing the method of claim 14, wherein said intercepting step includes detecting an attempt by the first process to open a communication channel to the second process.

As per claim 20: See col.5, lines 30-67 and col.12, lines 39-60; discussing the method of claim 14, wherein said intercepting step includes rerouting attempts by the first process to communicate with the second process from a system dispatch table to an interprocess communication controller.

As per claim 21: See col.5, lines 30-67 and col.12, lines 39-60; discussing the method of claim 14, wherein said step of identifying the second process includes evaluating parameters of the attempt made by the first process to communicate with the second process.

As per claim 22: See col.5, lines 30-67 and col.12, lines 39-60; discussing the method of claim 14, wherein said policy specifies particular processes to be protected from communications made by other processes.

As per claim 23: See col.5, lines 30-67 and col.12, lines 39-60; discussing the method of claim 14, further comprising: providing for a process to be registered in order to be protected from communications made by other processes; and determining whether to allow the first process to communicate with the second process based, at least in part, upon determining whether the second process is registered.

As per claim 24: See col.5, lines 30-67 and col.12, lines 39-60; discussing the method of claim 23, wherein said determining step is based, at least in part, on the type of communication the first process is attempting with the second process.

As per claim 25:

Dugan discloses in a computer system operating under control of an operating system supporting interprocess communication, a method for controlling interprocess communications from one application to another, the method comprising:

registering a first application to be protected from interprocess communications of other applications; **(col.5, lines 62-67)**

detecting an attempt to access the first application using interprocess communication; **(col.2, line 65 – col.3, lines 21 and col.6, lines 5-9)**

identifying a second application that is attempting to access the first application using interprocess communication; and **(col.5, lines 30-67 and col.12, lines 39-60)**

rerouting the attempt to access the first application through an interprocess communication controller that determines whether to allow the attempt **(col.4, lines 34-52 and col.8, lines 1-52)**, based on rules **(col.3, lines 1-9 and col.10, lines 5-18)** indicating whether the second application may access the first application using interprocess communication. **(col.4, lines 56-65 and col.8, lines 53-67)**

Ablay discloses the service environment can be embodied within a stand alone computer (col.3, lines 20-28) and a Windows NT operating system was used to provide the service creation environment (col.4, lines 20-31). Ablay also discusses the computer operating system must provide interoperability between the higher-level applications and the underlying computer hardware (col.5, lines 30-35) based on rules (col.9, lines 14-40). Thus, shows the operating system of a computer system supports interprocess communication that invokes service according to an application rule in a computer

system. However, a secondary art is brought forth to show the well known interprocess methodology existed wherein a computer system operating under the operating system supporting an interprocess communication.

Johnson is brought forth proving that interprocess communication may occur in the operating system (UNIX) of a computer system was known even prior to his invention. Johnsons invention facilitates interprocess communication among processes located at different nodes of a network (col.1, line 58—col.2, line 3). The interprocess communication within the UNIX system is taught by M.J. Bach in 1986 so that his invention is an improvement to Bach's. Thus, obviously shows that (Bach's) interprocess communication is supported by an operating system within a computer system rather than at different nodes of a network of Johnson's improvement.

As per claim 26: See col.4, lines 34-52 and col.8, lines 53-67; discussing the method of claim 25, wherein said registering step includes supplying rules specifying particular communications from which the first application is to be protected.

As per claim 27: See col.3, lines 13-28 and col.8, lines 1-7 and 53-67; discussing the method of claim 26, wherein the interprocess communication controller determines whether to allow the attempt based, at least in part, upon the rules specifying particular communications from which the first application is to be protected.

As per claim 28: See col.3, lines 13-28 and col.8, lines 1-7 and 53-67; discussing the method of claim 25, wherein said detecting step includes intercepting operating system calls for accessing the first application.

As per claim 29: See col.3, lines 60-63 and col.6, lines 10-15; discussing the method

of claim 25, wherein said detecting step includes detecting a graphical device interface (GDI) message sent to the first application.

As per claim 30: See col.3, lines 13-28 and col.8, lines 1-7 and 53-67; discussing the method of claim 29, wherein said identifying step includes evaluating parameters of the message sent to the first application.

As per claim 31: See col.3, lines 13-28 and col.8, lines 1-7 and 53-67; discussing the method of claim 25, wherein said detecting step includes detecting an attempt to send keystroke data to a window of the first application.

As per claim 32: See col.3, lines 13-28 and col.8, lines 1-7 and 53-67; discussing the method of claim 25, wherein said detecting step includes detecting an attempt to send mouse movement data to a window of the first application.

As per claim 33: See col.5, lines 30-67 and col.12, lines 39-60; discussing the method of claim 25, wherein said rerouting step includes rerouting the attempt to access the first application from a system dispatch table to the interprocess communication controller.

As per claim 34: See col.5, lines 30-67 and col.12, lines 39-60; discussing the method of claim 25, wherein said rules indicating whether the second application may access the first application includes rules indicating particular types of communications which are allowed.

As per claim 35: See col.5, lines 30-67 and col.12, lines 39-60; discussing the method of claim 25, further comprising: if the interprocess communication controller

allows the attempt to access the first application, routing the attempt to the first application.

As per claim 36:

Ablay discloses a system for regulating interprocess communication between applications, the system comprising:

a computer having at least one process, said computer system operating under control of an operating system supporting interprocess communication

a policy specifying applications that are permitted to communicate with a first application using interprocess communication;

a module for detecting a second application attempting to communicate with the first application using interprocess communication; and **(col.2, line 65 – col.3, lines 21 and col.6, lines 5-9)**

an interprocess communication controller for identifying the second application attempting to communicate with the first application **(col.5, lines 30-67 and col.12, lines 39-60)** and determining whether to permit the communication based upon the identification of the second application **(col.3, lines 1-21 and col.10, lines 5-18)** and the policy specifying applications permitted to communicate with the first application. **(col.4, lines 34-52 and col.8, lines 1-52)**

Ablay discloses the service environment can be embodied within a stand alone computer (col.3, lines 20-28) and a Windows NT operating system was used to provide the service creation environment (col.4, lines 20-31). Ablay also discusses the computer operating system must provide interoperability between the higher-level applications

and the underlying computer hardware (col.5, lines 30-35) based on rules (col.9, lines 14-40). Thus, shows the operating system of a computer system supports interprocess communication that invokes service according to an application rule in a computer system. However, a secondary art is brought forth to show the well known interprocess methodology existed wherein a computer system operating under the operating system supporting an interprocess communication.

Johnson is brought forth proving that interprocess communication may occur in the operating system (UNIX) of a computer system was known even prior to his invention. Johnsons invention facilitates interprocess communication among processes located at different nodes of a network (col.1, line 58—col.2, line 3). The interprocess communication within the UNIX system is taught by M.J. Bach in 1986 so that his invention is an improvement to Bach's. Thus, obviously shows that (Bach's) interprocess communication is supported by an operating system within a computer system rather than at different nodes of a network of Johnson's improvement.

As per claim 37: See col.3, lines 13-28 and col.8, lines 1-7 and 53-67; discussing the system of claim 36, wherein said policy includes rules indicating particular types of communications which are permitted.

As per claim 38: See col.3, lines 13-28 and col.8, lines 1-7 and 53-67; discussing the system of claim 36, further comprising: a rules engine for specifying applications that are permitted to communicate with the first application using interprocess communication.

As per claim 39: See col.4, lines 32-65 and col.5, lines 50-65; discussing the system

of claim 36, further comprising: a registration module for establishing said policy.

As per claim 40: See col.4, lines 32-65 and col.5, lines 50-65; discussing the system of claim 39, wherein said registration module provides for identifying applications to be governed by said policy.

As per claim 41: See col.3, lines 1-21 and col.5, lines 50-65; discussing the system of claim 36, wherein said module for detecting a second application detects an operating system call to open a communication channel to the first application.

As per claim 42: See col.3, lines 1-21 and col.12, lines 39-60; discussing the system of claim 36, wherein said module for detecting a second application detects a graphical device interface (GDI) message sent to the first application.

As per claim 43: See col.3, lines 1-21 and col.5, lines 50-65; discussing the system of claim 36, wherein said module for detecting a second application detects a local procedure call attempting to access the first application.

As per claim 44: See col.3, lines 1-21 and col.8, lines 1-7 and 53-67; discussing the system of claim 36, wherein said module for detecting a second application redirects attempts to communicate with the first application to the interprocess communication controller.

As per claim 45: See col.3, lines 13-28 and col.8, lines 1-7 and 53-67; discussing the system of claim 36, wherein said module for detecting a second application reroutes the attempt to communicate with the first application from a dispatch table to the interprocess communication controller.

As per claim 46: See col.3, lines 13-28 and col.8, lines 1-7 and 53-67; discussing the

system of claim 36, wherein said interprocess communication controller determines whether to permit the communication based, at least in part, upon evaluating parameters of the attempt made by the second application to communicate with the first application.

As per claim 47: See col.5, lines 30-67 and col.12, lines 39-60; discussing the system of claim 36, wherein said interprocess communication controller determines whether to permit the communication based upon obtaining user input as to whether to permit the second application to communicate with the first application.

Conclusion

4. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Leynna T. Truvan whose telephone number is (571) 272-3851. The examiner can normally be reached on Monday - Thursday (7:00 - 5:00PM).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kim Vu can be reached on (571) 272-3859. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/L. T. T./
Examiner, Art Unit 2135
/KimYen Vu/
Supervisory Patent Examiner, Art Unit 2135